Introducing Mathematica:
a system for doing mathematics by computer.

Department of Mathematics
Royal Holloway, University of London

I. INTRODUCTION

Mathematica is a powerful software package that, as the name suggests, allows you to do mathematics on a computer. It may be used as a sophisticated calculator, as we'll be doing here, or as a programming language. In the latter mode it contains the main elements from several other programming languages but is specifically designed to perform mathematical tasks easily. Just a few of its capabilities are illustrated here. Some of the examples involve topics like complex numbers and partial fractions that you may not have met before. For now, just type in what is given: you will probably learn about them later.
This document is also available online from the Student pages of the Departmental website; click through to 'Teaching material' and click on 'Introduction to Mathematica software'.

II. ENTERING AND EDITING COMMANDS

You interact with Mathematica by entering commands, and asking it to execute them. To execute a Mathematica command you must first type it on a new line, and then press the `Enter` key (the key labeled Enter on the numerical keypad at the extreme bottom right of the keyboard). Note that this is not the same key as the `Return` key, which is normally used to enter a new line: if you use that key you will just get a new line. If you are using a laptop, there is no Enter key, you need to press `Shift-Return` to execute a command; this also works on a full keyboard.
Let us begin with a simple command: adding 2 and 2. Type the following line followed by `Enter`. (Here and henceforth, the lines you have to type in are indicated by the symbol $\boxed{>}$.)

$\boxed{>}$ `2+2`

When you press `Enter`, you'll see that the display changes slightly, to

```
In[1]:= 2 + 2

Out[1]= 4
```

The first line is called the Input line, the second is the Output line.

Here's a more complicated command, and a typo has deliberately been made.

> D[Log[x+Arcsin[x]^2],x]

If you press Enter, you get a warning message:

```
General::"spell1": Possible spelling error: new symbol name "Arcsin"
is similar to existing symbol "ArcSin".
```

The typo is the 's' in 'Arcsin': it has to be a capital 'S'. Mathematica is case-sensitive!

Fortunately, you don't need to retype everything again. You can just position the cursor on the input line that you wish to correct, in this case at the 's', and make modifications in the usual way. Then press Enter again, and you'll get the correct answer.

You can also use Copy-and-Paste to reuse parts of previously entered commands. In short, Mathematica has all the functionality you'd come to expect from a word processor.

III. NUMERICS

Let's do some simple maths now. The arithmetic operators are the usual + and −, together with * for multiplication, / for division and ^ for power.

A. Rational Numbers

Try evaluating

> 1/3 + 1/6 and

> 3*2^4/7

Notice that division is done before addition and the power is calculated before division. The answer in each case is output as a rational number. We say that Mathematica uses *exact arithmetic*.

Round brackets are used to change the order in which operations are performed. Enter

> 1/(3 + 1/6)

B. Decimals

Decimal approximations may be obtained using the N function. We always use square brackets to enclose the parameters of a function. Try

> N[3*2^4/7]

This gives only 6 significant figures. This may be increased to any number of significant figures by giving a second parameter to the function. The parameters are separated by a comma ,

> N[3*2^4/7,50]

If you include a decimal point somewhere in a formula, as in

> 3*2.^4/7

Mathematica takes that as a hint to use decimal numbers and the output will indeed be shown as a decimal number.

The accuracy of Mathematica is only limited by the speed and available memory of the computer. You want 20,000 significant digits? You can get them (eventually)!

C. Large Integers

It is also possible to calculate very large integers. Try

> 2^100

> 50!

D. Irrationals

The word `Pi` is the Mathematica symbol for π. The capital P is important; in Mathematica the names of all built-in symbols and functions begin with a capital letter. If you enter `Pi` it will just be echoed on the next line. To find an approximate numerical value of π enter

> N[Pi]

which should give the result 3.14159, with the default 6 significant figures. Exercise: try and get π to 10 decimal places, 20 and 1000.

The square root of 2 can be entered either as `2^(1/2)` or `Sqrt[2]`, but in both cases no decimal approximation will be made. Note the round brackets: they are necessary to enclose the exponent $1/2$. Check what happens if the brackets are omitted. To obtain the value to 6 decimals enter

> N[Sqrt[2],7]

An alternative way to obtain a decimal approximation is to follow the 2 with a decimal point, thus `Sqrt[2.]`, which gives simply the default 6 significant figures.

E. Palettes

Symbols such as $\sqrt{\cdot}$ and π may also be typed using the BasicMathInput Palette. If this Palette is hidden, left-click *Palettes* on the Toolbar and select *BasicMathInput*. Left-Click to select the required

symbol. If necessary, either click the left mouse ear or use the `Tab` key to jump to the various placeholders within a symbol.

Use the Palette to evaluate

> `N[3/19]` and

> $\sqrt[3]{8}$

F. Logarithms

The function `Log[x]` calculates the natural logarithm of x, usually denoted by $\ln x$. The base of the natural logarithm, e, is denoted in Mathematica by `E` and is treated as an exact irrational number in the same way as π. Try entering

> `Log[E]`

> `E^Log[1/2]`

Also obtain the square root of e to three decimal places. To calculate logarithms with a different base, the base must be entered as the first parameter of the `Log` function. Enter the following and then obtain a decimal approximation to the second expression by including a decimal point after the 10.

> `Log[2,8]` and

> `Log[3,10]`

G. Different number systems

The base of a number may be changed using `BaseForm`. For example, 6 in base 2, i.e. the binary representation of 6, is given by

> `BaseForm[6,2]`

H. Complex Numbers

The square root of -1 is represented by `I`, or the i symbol on the Palette. Confirm this by entering

> `I^2`

> `1/I`

Also try the following

> `(2 + I)^3`

> `(1 + I)(1 - I)`

> `1/(1 + I)`

Now use Mathematica to find the square root of -1 and the logarithm of -1.

I. Separating variables

When you are multiplying two variables, you need either a space or a `*` to separate them: `x y` or `x*y` will do, but `xy` will be interpreted as a new variable name. You don't need to separate numbers from variables, though. E.g. `5x` and `7y^3` are fine.

IV. FUNCTIONS AND GRAPHICS

A. Plotting functions

Mathematica contains all of the functions contained on a standard calculator and many more. The nature of a function may be visualised by plotting its graph. This is done with the Mathematica function `Plot`. To see the parabola $y = x^2$ for x between -2 and 2, type

> `Plot[x^2, {x,-2,2}]`

The function is specified first, followed by the range over which you wish to plot it, in the format {variable , start value , end value}.

To see four cycles of the periodic function $\sin x$ enter

> `Plot[Sin[x], {x,-4Pi,4Pi}]`

The sine function is written as `Sin[x]` because it is a built-in Mathematica function name. Make sure to type capital S. If you type `Plot[sin[x], {x,-4Pi,4Pi}]` (with lower case s) you will not get a plot of the sine function because Mathematica does not recognise `sin` as its sine function. Also, the arguments of a Mathematica function have to be surrounded by square brackets `[]`, not round ones `()`, nor curly ones `{}`.

A periodic function which you may not have met before is `Mod[x,p]`, which gives the remainder when x is divided by p. The graph of this function is a sawtooth function having period p. Try

> `Plot[Mod[x,2],{x,-4,4}]`

Inverse trigonometric functions are denoted by the prefix `Arc`; for example, the inverse of the sine function is called `ArcSin`. Again, mind the capital A and S. To plot the inverse sine function,

> `Plot[ArcSin[x],{x,-2,2}]`

Notice that the x-interval has been truncated to the interval $[-1, 1]$. Why is this? If you know the behaviour of the sine function it should be pretty clear.

Now try `ArcTan`. You can copy and edit the previous line.

The absolute value $|x|$ of a number x is entered as `Abs[x]`. Use this to plot the function $|x|/x$ between $x = -3$ and $x = 3$.

B. Define your own functions

It is possible to define your own functions. For example, you can define a function called 'parabola' by entering

```
> parabola[x_]:=(x-1)^2
```

Mind the underscore `_` right after the first x. This tells Mathematica that you will use x as the argument in function definition.

Now you won't see any output. The line has had an effect, though, because Mathematica will now know that whenever you use the `parabola[x]`, you actually mean $(x-1)^2$. Indeed, when you enter

```
> Plot[parabola[x], {x,0,2}]
```

you will see the graph of the function `parabola`, which is a parabola with vertex at $(1, 0)$.

Note that it is a good idea always to use small letters in functions that you define yourself, so that they don't get confused with built-in Mathematica function names.

C. Composition of functions

Functions may be composed in the usual way. A function of period 2 which is parabolic in each cycle may be graphed by

```
> Plot[parabola[Mod[x,2]],{x,-4,4}]
```

D. Plotting functions of 2 variables

The graph of a function of two variables is a surface; enter

```
> Plot3D[x^2+y^2,{x,-2,2}, {y,-4,4}]
```

This is a paraboloid of revolution. Changing the sign of the y^2 term gives a saddle:

```
> Plot3D[x^2-y^2,{x,-2,2},{y,-4,4}]
```

A function which is periodic in both variables is obtained by

```
> Plot3D[parabola[Mod[x,2]]+parabola[Mod[y,2]], {x,-2,2},{y,-4,4}]
```

Note that you can manipulate a 3D image using the mouse pointer. You can enlarge images, and you can rotate the surfaces to get a view from any direction.

V. ALGEBRA

A. Factorisation and roots of a polynomial

The function `Factor` will factorise a polynomial; enter

```
> Factor[2-3x+x^2]
```

The function `Expand` has the opposite effect

```
> Expand[(x-1)(x-2)(x-3)]
```

Now try

```
> Factor[2 -4x +x^2]
```

Since the coefficients were integers Mathematica assumes that you are looking for factors with integer coefficients. Factorisation in terms of real numbers is obtained in this example by typing

```
> Factor[2. -4x +x^2]
```

since $2.$ is interpreted as a real number.

The roots of the quadratic are found exactly using

```
> Solve[2 -4x +x^2==0,x]
```

Note that equations need double equal signs. A single equal sign indicates an assignment, as in $x = 2$.

You can solve equations numerically using `NSolve`

```
> NSolve[2 -4x +x^2==0,x]
```

Notice how the roots are related to the result of `Factor`. Plot a graph of $2 - 4x + x^2$ for $0 \leq x \leq 4$ and notice that it cuts the x-axis where expected.

`NSolve` is not restricted to finding the roots of a quadratic: use it to find the roots of $7 - 12x + 7x^2 - x^3$. Notice that two of the roots are complex but have a real part near $x = 1$. Plot the graph of this cubic for x in the interval $[0, 6]$ to see what is happening. Now find the roots when the constant term 7 is replaced by 5 and plot the graph (copy and edit the previous lines to save typing).

Mathematica always attempts to find exact solutions. Compare this with the results of

```
> Solve[7-12x+7x^2-x^3==0,x]
```

B. Partial fractions

Partial fractions are formed using the function `Apart`; for example enter

```
> Apart[(2+x^2)/((1+x^2)(1-x)(1-2x))]
```

So that we can refer to the result of this calculation again we save it by assigning it to a new variable called '`fraction`'

```
> fraction=Apart[(2+x^2)/((1+x^2)(1-x)(1-2x))]
```

and then we can put everything back together over a common denominator by

```
> Together[fraction]
```

VI. CALCULUS

A. Derivatives and Integrals

To differentiate $\tan x$ enter

```
> D[Tan[x],x]
```

Now differentiate $\log(\sin x)$, remembering when you need capital letters and square brackets.

A more complicated example is

```
> D[Log[x+ArcSin[x]^2],x]
```

Indefinite integrals can often be done in terms of simple functions: try

```
> integral = Integrate[x*Sin[x]^2,x]
```

This saves the result of the integration under the name '`integral`'. The result may be checked by differentiation

```
> D[integral,x]
```

The result of the differentiation should be the integrand but this is not always explicitly so without further manipulation. If necessary, enter

```
> Simplify[%]
```

The function `Simplify` reorganises an expression into a 'simpler' form. In this case the expression is the symbol %, which stands for the previous line of output.

Now enter

```
> int=Integrate[(1+2x+3x^2)/(1+x)^3,x]
```

and notice where the three terms come from by inspection of the partial fraction above. Check the result using

```
> der=D[int,x]
```

and either simplify the output or collect the terms together.

Most indefinite integrals which can be expressed in terms of standard functions can be done by Mathematica but sometimes the functions may be new to you.

```
> Integrate[1/Sqrt[1+x^2],x]
```

(`ArcSinh` is the inverse hyperbolic sine function, which will become familiar, if it isn't already.)

Sometimes the integral can be difficult to find by hand: look at

```
> Integrate[1/(x^4+1),x]
```

and check that the derivative of the integral is the original function (you will need to use `Simplify`).

For definite integrals (integrals with limits), consider this example: $\int_{-1}^{4}(x+1)^3dx$ is entered as

```
> Integrate[(x+1)^3,{x,-1,4}]
```

Consider this:

```
> f = a*x^2 + b
```

```
> Integrate[f,x]
```

```
> Integrate[f,a]
```

What do you think `Integrate[f,b]` will be?

A couple more integrals to try:

$$\int \frac{x}{\sqrt{16-x^4}}dx \quad \text{and} \quad \int_1^3 |\ln(x/2)|dx.$$

In each case think about how you might try them without the use of Mathematica.

A fact to remember is that Mathematica does not include the constant of integration in an indefinite integral: you have to add it yourself. Sometimes Mathematica does not give you the answer you expect, because of this missing constant. You should always check that apparently different answers are nothing more than that – apparently different – by doing the necessary calculations.

B. Lists

A list is a collection of elements separated by commas and enclosed in curly brackets. Let

```
> L1 = {1,2,3}
```

```
> L2 = {2,3,4}
```

A list may be multiplied by a constant; try

```
> 2*L1
```

If addition is defined for the elements and the lists are of the same length then they can be added together like vectors. Enter

```
> L2 + 2*L1
```

Here, `L1*L2` calculates the pairwise product of the elements, which is not a vector operation. The scalar product of two vectors is instead given by `L1.L2`. To compare the results, enter

```
> {x1,y1,z1,}.{x2,y2,z2}
> {x1,y1,z1}*{x2,y2,z2}
```

Now enter

```
> L1*L2
> L1.L2
```

and check the results.

A very useful Mathematica function for making a list of values of a function is `Table`. Enter the following two examples

```
> L3 = Table[x^2, {x, 0, 10}]
> L4 = Table[N[Sin[x]],{x,0,Pi,Pi/10}]
```

The last parameter defines the intervals at which $\sin x$ is to be evaluated; without it, the function is evaluated at consecutive integers, as in `L3`. It is interesting to run the `L4` command without the `N` function.

VII. SOLVING EQUATIONS

A. Solving non-polynomial equations

By definition, arcsinh(25) is the value of x such that $\sinh x = 25$. The value of x may be found using the Mathematica function `FindRoot`; enter

```
> FindRoot[Sinh[x]==25, {x,4}]
```

Note the == sign again. Check the result by entering

```
> N[ArcSinh[25]]
```

`FindRoot` uses an iterative method which determines a sequence of closer and closer approximations to a solution of the given equation. The parameter 4 sets the first point $x = 4$ for the iteration.

B. Solving simultaneous equations

Simultaneous equations can be solved by entering a list of equations followed by a list of variables to be solved for as parameters to the `Solve` function (or `NSolve`). The following solves a pair of linear equations.

```
> Solve[{x+y==13,2x+7y==5},{x,y}]
```

Introducing a third variable z and no further equations means that x and y will now be functions of z:

```
> Solve[{x+y+3z==13,2x+7y-z==5},{x,y}]
```

The equations to solve are not necessarily linear:

```
> Solve[{x^2+y^2==2,x+y==1},{x,y}]
```

C. Solving differential equations

This is similar in form to solving an algebraic equation, except that we are solving for a function such as y of the variable x, i.e. $y(x)$, rather than the usual variable x. For example

```
> DSolve[y''[x] + 5y'[x] + 4y[x] == Exp[x], y[x], x]
```

(Use a double dash for the second derivative, rather than quotation marks.) The answer will involve arbitrary constants, if no boundary conditions are given. To include boundary conditions, they appear as part of the list of equations to be solved, as in

```
> DSolve[{y''[x]+5y'[x]+4y[x]==Exp[x],y[0]==1,y'[0]==0}, y[x],x]
```

VIII. MORE GRAPHICS

A. Parametric Plots

Another useful plotting function is `ParametricPlot`, which as its name suggests is for plotting parametric curves. Enter

```
> ParametricPlot[{t,t^2},{t,-2,2}]
```

which plots the parabola defined by the parametric equations $x = t$, $y = t^2$. Now try

```
> ParametricPlot[{Cos[t],Sin[2t]},{t,0,2Pi}]
```

B. Polar Plots

Sometimes it is useful to define a curve using polar coordinates (r, θ) in terms of which $x = r\cos\theta$ and $y = r\sin\theta$. Giving r as a function of θ then defines x and y as functions of θ and hence gives a parametrically defined curve with θ as parameter. Thus, for example, $r = \sin\theta$ gives $x = \sin\theta\cos\theta$ and $y = \sin^2\theta$ which for θ in the interval $[0, \pi)$ should give a circle. Replacing θ by `t` in `ParametricPlot` results in the code

```
> ParametricPlot[Sin[t]{Cos[t],Sin[t]},{t,0,Pi}]
```

Notice that `Sin[t]` has been factored out of the list of functions.

Plot $r = 1/(1 + 0.5\cos\theta)$.

C. Selecting the plotting range

The option `PlotRange` allows you to specify the range of x and y values to be displayed in the graph. The use of this is illustrated by the following example – in the second line below the range of y-values is specified as shown. Compare the two graphs.

```
> Plot[Sin[x]/x,{x,0,10Pi}]
> Plot[Sin[x]/x,{x,0,10Pi},PlotRange->{-1,1}]
```

D. Plotting Points

The function `ListPlot` enables a set of points to be plotted and optionally connected by straight lines. Define the list of squares by `L3`, from above, and then enter

```
> ListPlot[L3, PlotJoined->True, AxesOrigin->{0,0}]
```

The second option insists that the origin of the axes is $(0, 0)$. Notice that the first point is plotted at $x = 1$. The default for the x-coordinates when not specified is $1, 2, 3, 4, \ldots$ The x-coordinates may be specified as in the following example which should draw a hexagon by joining together points on the unit circle.

```
> L5 = Table[{Cos[t],Sin[t]},{t,0,2Pi,Pi/3}]
> ListPlot[L5, PlotJoined->True]
```

Enter

```
> L5
```

to see the structure of `L5`: it is a list of pairs of x-y values.

E. Superimposing more than one graph

Suppose, for example, we want to superimpose a function f and its derivative. Let the function be defined by

```
> f[x_]:=Sin[x]
```

Then the superposition is achieved by

```
> Plot[{f[x],f'[x]},{x,0,2Pi}]
```

The colours in which the curves are drawn can be varied using the `PlotStyle` option.

```
> Plot[{f[x],f'[x]},{x,0,2Pi},PlotStyle->{Hue[0.0],Hue[0.9]}]
```

Sometimes we need to superimpose graphs which are defined on different intervals. When two intervals are involved, the easiest command to use is the `If` command. Try

```
> f[x_]:=If[x<=1, x, x-2]
```

```
> Plot[f[x], {x, -1,3}]
```

When more than two intervals are involved, it is easier to use the `Which` command, illustrated in

```
> g[x_]:=Which[x<=-1,1,-1<=x<=1,-x,x>1,-1]
```

```
> Plot[g[x],{x,-3,3}]
```

IX. MORE CALCULUS

A. Higher Order Derivatives

The second derivative of $\tan x$ is found by using

```
> D[Tan[x],{x,2}]
```

Now use Mathematica to find the third derivative of $\sin^3 x$.

B. Partial Derivatives

In the case of a function of two variables, x and y say, the partial derivative with respect to x is the derivative treating y as a constant (remember the spaces when multiplying here!).

```
> D[Sin[x y]/(x y^2), x]
```

Find the partial derivative with respect to y.

The second x-derivative is given by

```
> D[Sin[x y]/(x y^2), {x,2}]
```

Finally the cross-derivative is obtained by introducing a third argument

```
> D[Sin[x y]/(x y^2), {x,1},{y,1}]
```

X. FURTHER PARAMETRIC PLOTS

A. Lissajous figures

The family of Lissajous figures has two parameters, m and n. Enter the definition

```
> Clear[g];
> g[m_,n_]:=ParametricPlot[{Cos[m*t],Sin[n*t]},{t,0,2*Pi }]
```

The circle is a special case which can now be displayed with `g[1,1]`.

Now try `g[5,4]` and `g[14.5,15]` and a few other values of your choice, time permitting.

B. The Sierpinski gasket

The Sierpinski gasket is a famous fractal object (you will learn all about this object in the course MT1100 From Euclid to Mandelbrot). Try entering this:

```
> left={0,0}; right={1,0}; top={0.5,0.5*Sqrt[3]};
```

Check that this gives you the vertices of an equilateral triangle; ignore warnings about similar spellings, but this time it is essential that you use lower case letters where shown.

```
> f[x_]:=0.5*(x+Switch[Random[Integer,{1,3}],1,left,2,right,3,top])
```

This gives you a point in the triangle halfway between the last point and a vertex chosen at random.

```
> data=NestList[f,{0,0},50000];
```

The semicolon at the end *is* important here if you want to finish today. It suppresses any output to screen; in this case it is a huge list of numbers that would take ages to display.

```
> ListPlot[data,PlotStyle->PointSize[0.001]]
```

Not having the semicolon is important here.

What's going on here? `left`, `right` and `top` are the vertices of an equilateral triangle. The next instruction takes one point x and finds another point $f(x)$ which is halfway between x and one of the vertices chosen at random. Then $f(x)$ replaces x. The last instruction plots 50000 successive values of x. This gives the gasket, a fractal of dimension $\ln 3/\ln 2 = 1.58\ldots$

To get a feel for why, draw an equilateral triangle and join the midpoints of the sides to each other to make four smaller triangles. Now consider any point in the original triangle. The midpoint of that point and a vertex must lie in the smaller triangle nearest the vertex, and will never lie in the

central smaller triangle. So that central triangle cannot contain any points after the first point. At the next stage there can be no points in the central triangle of the three smaller ones and so on.

XI. THE NATIONAL LOTTERY

We will now use random numbers to simulate the National Lottery. First make a list of ball numbers by entering

```
> balls = Range[1,49]
```

Now choose your numbers by assigning a list of six ball numbers to the variable `ticket` as in the following example, but please choose your own numbers.

```
> ticket = {1,4,21,30,32,33}
```

To play the game enter the following code which selects the winning ball numbers. Be careful to include the colon in `:=` and only enter the code after typing all of the lines. Use the return key to move to the next line. No output is expected.

```
> play := (balls=Range[1,49];
> Do[b = Random[Integer,{1,Length[balls]}];
> B[i] = balls[[b]];
> balls = Delete[balls,b],{i,1,6}];
> {B[1],B[2],B[3],B[4],B[5],B[6]})
```

Each occurrence of the variable play will be replaced by a random list of six different ball numbers. Different numbers are assured since the code deletes the chosen ball before selecting the next one. So try entering

```
> p=play
```

The Mathematica instruction `Intersection` forms the intersection of two sets so in order to determine which of your chosen numbers were winners enter

```
> Intersection[p,ticket]
```

Combining these lines gives a function which plays again and returns only the winning numbers which appear on your ticket

```
> winners:=Intersection[play,ticket]
```

Try entering the word winners several times. To get the results of playing fifty times, enter

```
> t=Table[winners,{50}]
```

To select the games for which you had two winners enter

```
> Select[t,Length[Slot[#1]]==2&]
```

The number of games with two winners is

```
> win2 = Length[Select[t,Length[Slot[#1]]==2&]]
```

A. Theory

The Lottery selects 6 balls from 49 and the total number of such selections is the binomial coefficient

$$\frac{49!}{43!6!} = 13983816.$$

To check it,

```
> total = Binomial[49,6]
```

So the probability of winning the jackpot is about 1 in 14 million. The number of ways in which exactly n of the numbers on your ticket will be selected is the number of ways of selecting n of the 6 numbers on your ticket and $6 - n$ of the remaining 43 numbers. The probability that a given ticket will have n winning numbers is therefore determined by the function

```
> pr[i_]:= N[Binomial[6,n]*Binomial[43,6-n]/total, 3]
```

and a list of these probabilities is obtained by entering

```
> Table[pr[n],{n,0,6}]
```

which should give $(0.436, 0.413, 0.132, 0.0177, 0.000969, 0.0000184, 7.15 \times 10^{-8})$, which is to be compared with the simulated result above. Notice that 85% of the time you expect to get no more than one winner. A person who buys one ticket a week expects to get three winners about once a year.

XII. IF THINGS DON'T WORK

The most likely problems are:

- using the wrong sort of bracket: round brackets for grouping, square brackets in functions, and curly brackets for lists or ranges;
- forgetting to use * or a space to separate two variables in a product;
- forgetting the square brackets in a function;
- forgetting that built-in Mathematica functions need a capital letter.